International Journal of Computer Networks and Distributed Computing
Vol. 4, Issue 1

A New Distributed Web Page Classification

Moeli Reapel[1], R.K Kloperizoty [2]

Master of Technology, Dept. of Information Technology, U V Patel College of Engineering, Kherva, Mehsana, Gujarat, India[1]

Assistance Professor, Dept. of Information Technology, U V Patel College of Engineering, Kherva, Mehsana, Gujarat, India[2]

Abtract: Web page classification is the process of classifying documents into predefined categories based on their content. The task of data mining is to automatically classify documents into predefined classes based on their content. Many algorithms have been developed to deal with automatic text classification. The most common techniques used for this purpose work include Apriori Algorithm and implementation of Naive Bayes Classifier. Apriori Algorithm finds interesting association or correlation relationships among a large set of data items. The discovery of these relationships among huge amounts of transaction records can help in many decision making process. The Naive Bayes Classifier uses the maximum a posterior estimation for learning a classifier. Then, use Naive Bayes Classifier to calculate probability of keywords among a large data itemsets. Moreover, this technique is efficient for web page classification. The technique will be more effective is the training set is set in such a way that it generates more sets. Though the experimental results are quite encouraging, it would better if the work with larger data sets with more classes.

Keywords: Association Rule, Apriori Algorithm, Naïve Bayes Classifier

I. INTRODUCTION

There are numerous web pages available in electronic form. Such documents represent a massive amount of information that is easily accessible. Seeking value in this huge collection requires organization; much of the work of organizing documents can be automated through data mining [4][11]. The accuracy and our understanding of such systems greatly influence their usefulness. The task of web mining is to automatically classify documents into predefined classes based on their content. Many algorithms [5] have been developed to deal with automatic text classification [5]. The most common techniques used for this purpose include Apriori Algorithm [5][6][8] and implementation of Naive Bayes Classifier [5][6].

Apriori Algorithm [5][6] finds interesting association[2][8] or correlation relationships among a large set of data items. The discovery of these relationships among huge amounts of transaction records can help in many decision making process. On the other hand, the Naive Bayes Classifier [5][6] uses the maximum a posterior estimation for learning a classifier. Then, the Naive Bayes Classifier [5][6] to calculate probability of keywords among a large data itemsets.

II. BACKGROUND STUDY

A. Data Cleaning [7][11]

Data cleaning [7], also called data cleansing or scrubbing, deals with detecting and removing errors and inconsistencies from data in order to improve the quality of data. Data quality problems are present in single data collections, such as files and databases, e.g., due to misspellings during data entry, missing information or other invalid data. When multiple data sources need to be integrated, e.g., in data warehouses, federated database systems or global web-based information systems, the need for data cleaning increases significantly.

Most natural languages have so-called function word and connections such as articles and preposition that appear in a larger number if documents and are typically of little use in pinpointing documents that satisfy a searcher's information need. Such words (e.g., a, an, the, on for English) are stop words.
Stop words - words which do not contain important significant information or occur so often that in text that they lose their usefulness.

Following steps to remove noisy data in each web page:

- In this research, first remove the noisy data by using application.
- Each abstracts used to train is considered as a transaction in the text data.
- The text data is cleaned by removing unnecessary words or noisy data i.e. text data is filtered and subject related words are collected.

Therefore, a useful pre-processing step is to run your data through some data cleaning [11] routines.

B. Association Rule [12]

Association rules are an important class of regularities in data. Mining of association rules is a fundamental data mining task. It is perhaps the most important model invented and extensively studied by the database and data mining community. Its objective is to find all co-occurrence relationships, called associations, among data items.

The left hand side of an association rule is called the antecedent, and the tight hand side is the consequent. In the Cheese → Beer example Beer is the antecedent and Cheese is the consequent.

The classic application of association rule mining is the market basket data analysis, which aims to discover how items purchased by customers in a supermarket (or a store) are associated. An example association rule is

Cheese → Beer [support = 10%, confidence = 80%]

The rule says that 10% customers buy Cheese and Beer together, and those who buy Cheese also buy Beer 80% of the time.

The problem of mining association rules can be stated as follows: Let $I = \{i1, i2, \ldots, im\}$ be a set of items. Let $T = (t1, t2, \ldots, tn)$ be a set of transactions (the database), where each transaction *ti* is a set of items such that *ti subset ofI*. An association rule is an implication of the form,

$X \rightarrow Y$, where $X \subset I, Y \subset I$, and $X \cap Y = \emptyset$

X (or Y) is a set of items, called an itemset.

Support: The support is the ratio (or percentage) of the number of itemsets satisfying both antecesent and consequent to the total number of transaction [9]. The support of a rule, $X \rightarrow Y$, is the percentage of transactions in T that contains$X \cup Y$, and can be seen as an estimate of the probability, $\Pr(X \cup Y)$. The rule support thus determines how frequent the rule is applicable in the transaction set T. Let n be the number of transactions in T. The support of the rule $X \rightarrow Y$ is computed as follows:

$$support = \frac{(X \cup Y).count}{n} \quad \ldots (1)$$

Support is a useful measure because if it is too low, the rule may just occur due to chance. Furthermore, in a business environment, a rule covering too few cases (or transactions) may not be useful because it does not make business sense to act on such a rule (not profitable).

Confidence: Confidence (strength or evidence) is derives from a subset of the transaction in which two entities (or activities) are related [9]. The confidence of a rule, $X \rightarrow Y$, is the percentage of transactions in T that contain X also contain Y. It can be seen as an estimate of the conditional probability, $Pr(Y \mid X)$. It is computed as follows:

$$confidence = \frac{(X \cup Y).count}{X.count} \quad \ldots (2)$$

Confidence thus determines the predictability of the rule. If the confidence of a rule is too low, one cannot reliably infer or predict Y from X. A rule with low predictability is of limited use.

C. Apriori Algorithm [5][6][8]

Apriori [5][6][8] is a strongly influencing later development algorithm for finding frequent itemsets using candidate generation. Apriori [5][6][8] is an influential algorithm for mining frequent itemsets for Boolean association rules. The name of the algorithm is based on the fact that the algorithm uses prior knowledge of frequent itemsets properties.

Apriori [5][6][8] employs an iterative approach known as a level-wise search, where *k*-itemsets are used to explore *(k+1)*-itemsets. First, the set of frequent 1-itemsets is found. This set is denoted by L_1. L_1 is used to find L_2, the set of frequent 2-itemsets, which is used to find $L3$, and so on, until no more frequent *k*-itemsets can be found. The finding of each L_k requires one full scan of the database.

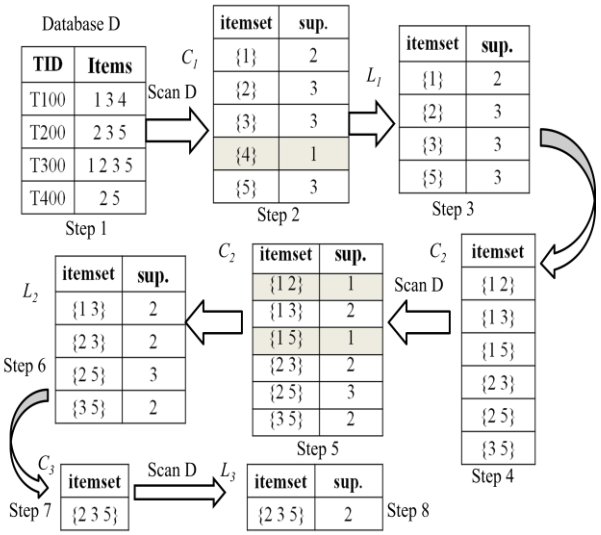

Fig. 1 Example of Apriori Algorithm

- In the first iteration of the algorithm, each item is a number of the set of candidate 1-itemsets, *C1*. The algorithm simply scans all of the transactions in order to count the number of occurrences of each item.
- Suppose that the minimum transaction support count required is 2 (i.e.; min_sup = 2/5 = 40%). The set of frequent 1-itemsets, *L1*, can then be determined. It consists of the candidate 1-itemsets satisfying minimum support.
- To discover the set of frequent 2-itemsets, *L2*, the algorithm uses *L1* | *L2* to generate a candidate set of 2-itemsets, *C2*.
- The transactions in *D* are scanned and the support count of each candidate itemset in *C2* is accumulated.
- The set of frequent 2-itemsets, *L2*, is then determined, consisting of those candidate-itemsets in *C2* having

minimum support.

- The generation of the set of candidate 3-itemsets, *C3* is observed in step 7 to step 8. Here *C3 = L1 | L2 = {{1, 2, 3}, {1, 2, 5}, {1, 3, 5}, {2, 3, 5}}*. Based on the Apriori property that all subsets of a frequent itemset must also be frequent, we can determine that the four latter candidates cannot possibly be frequent.
- The transactions in *D* are scanned in order to determine *L3*, consisting of those candidate 3-itemsets in *C3* having minimum support. The algorithm uses *L3 | L4* to generate a candidate set of 4-itemsets, *C4*. Although the join results in *{{1, 2, 3, 5}}*, this itemset is pruned since its subset *{{2, 3, 5}}* is not frequent. Thus, *C4 = {}*, and the algorithm terminates.

To understand how Apriori [5][6][8] property is used in the algorithm, let us look at how L_{k-1} is used to find L_k. A two step process is followed, consisting of join and prune actions:

i. The Join Step:

To find L_k, a set of candidate *k*-itemsets is generated by joining L_{k-1} with itself. This set of candidates is denoted by C_k. Let l_1 and l_2 be itemsets in L_{k-1} then l_1 and l_2 are joinable if their first *(k-2)* items are in common, i.e., $(l_1[1] = l_2[1])$. $(l_1[2] = l_2[2])$ . . . . . $(l_1[k-2] = l_2[k-2])$. $(l_1[k-1] < l_2[k-1])$.

ii. The Prune Step:

C_k is the superset of L_k. The scan of database to determine count, if each of candidate in C_k would result in the determination of L_k (itemsets having a count no less than minimum support in C_k). But this scan and computation can be reduced by applying the Apriori property. Any *(k-1)*-itemsets that is not frequent cannot be a subset of a frequent *k*-itemset. Hence if any *(k-1)*-subset of a candidate *k*-itemset is not in L_{k-1}, then the candidate cannot be frequent either and so can be removed from C_k.

The algorithm is as follows:

Input: Database, *D*; minimum support threshold, *min_sup*.

Output: *L*, frequent itemsets in *D*.

(1) L_1 = find frequent 1-itemsets(*D*);
(2) for(k= 2; $L_{k-1} \neq \emptyset$*; k++)*
(3) {
(4) C_k = apriori-gen(L_{k-1}, *min_sup*);
(5) for each transaction $t \in D$ //scan *D* for counts
(6) {
(7) *Ct = subset(Ck,t)*; //get the subsets of t that are candidates
(8) for each candidate $c \in Ct$
(9) *c.count++;*
(10) }
(11) Lk = {C ∈ Ck | c.count ≥ minimum_sup }
(12) }
(13) *return L=Uk Lk*;

The Apriori [5][6] achieves good performance by reducing the size of candidate sets. However, in situations with very many frequent itemsets, large itemsets, or very low minimum support, it still suffers from the cost of generating a huge number of candidate sets and scanning the database repeatedly to check a large set of candidate itemsets.

D. Naive Bayes Classifier [5][6][8][9]

Bayesian Classification [5][6][2] is based on Bayes theorem. A simple Bayesian Classification namely the Naive Classifier [5][6][8] is comparable in performance with decision tree [1][3] and neural network classifiers. Bayesian Classifiers [5][6][2] have also exhibited high accuracy and speed when applied to large database.

While applying Naive Bayes Classifier [5][6][8] to classify text, each word position in a document is defined as an attribute and the value of that attribute to be the word found in that position. The calculation of first term of this classifier is based on the fraction of each target class in the training data.

$$V_{NB} = argmax\, P(v_j) \Pi P(a_i/v_j) \quad \ldots (3)$$

Then the second term of the equation is calculated by the following equation after adopting m-estimate approach in order to avoid zero probability value,

$$\frac{n_k+1}{n+|vocabulary|} \quad \ldots (4)$$

where, *n* = Total no of word set position in all training examples whose target value is *j*, n_k = No. of times the word set found among all the training examples whose target value is *j*, |vocabulary| = The total number of distinct word set found within all the training data

"What if I encounter probability values is zero?" There is simple trick to avoid this problem. To assume that our training database, *D*, is so large that adding one to each count that to need would only make a negligible difference in the estimated probability value, yet would conveniently avoid the case of probability values of zero. This technique for Laplacian correction or Laplace estimator, named after Pierre Laplace, a French mathematician who lived from 1749 to 1827. If *q* counts to which to add one, then remember to add *q* to the corresponding denominator used in the probability calculation [9].

III. THE PROPOSED METHOD

The proposed method to classify text is an implementation of Apriori Algorithm. In this first collect the large data items on the electronic form. Then after remove the noise to using data cleaning techniques. Now implement the Apriori Algorithm and to find out the keywords of the data for all category related topics and obtain probability using Naive Bayes Classifier.

A. The Proposed Algorithm

The following algorithm is applicable at the class determination phase of testing phase. That is after the

probability table as well as association rules have been created for the training data, text preparation for the test data is done and then this algorithm is applied for the classification.

The propose algorithm is work as follows:

1. For each set no. of input files, min_sup, min_confi
2. To pre-process for each file
3. Obtain keywords
    4. If count/no. of input files is greater than min_sup then
    5. To save candidate sets and count
    6. Else to discard
    7. Calculate the probability for each set and each class
    8. End

B. Flow Chart of the Technique

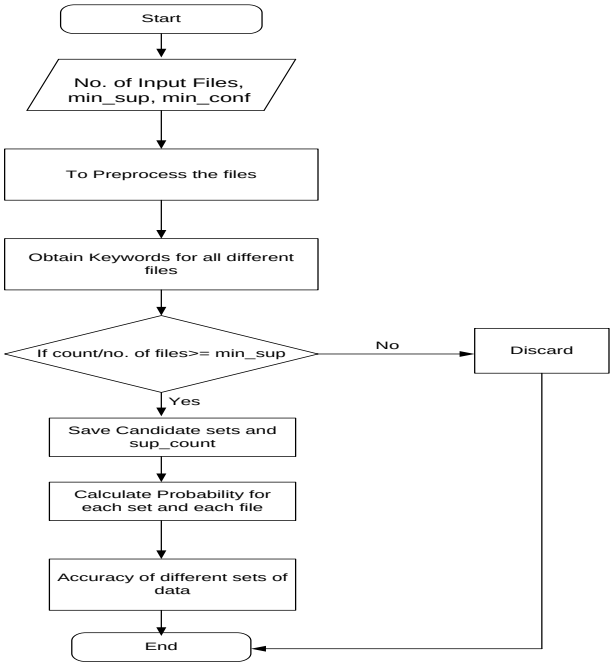

Fig. 2 Flowchart of the Proposed Algorithm

IV. EXPERIMENTAL EVALUATION

A. Preparing Webpage for Classification

To take webpage from different sites or different types have been used to analyze the experiment. Here i take five different classes are as Cricket, Hockey, Tennis, Football,Baseball. I collect number of web pages related to their different classes.

To make the raw text valuable, that is to prepare the text, considered only the keywords. That is unnecessary words and symbols are removed. For this keyword extraction process to drop the common unnecessary words like am, is, are, to, from .etc. and also dropped all kinds of punctuations and stop words. Singular and plural form of a word is considered same. Finally, the remaining frequent words are considered as keywords.

Let web page:

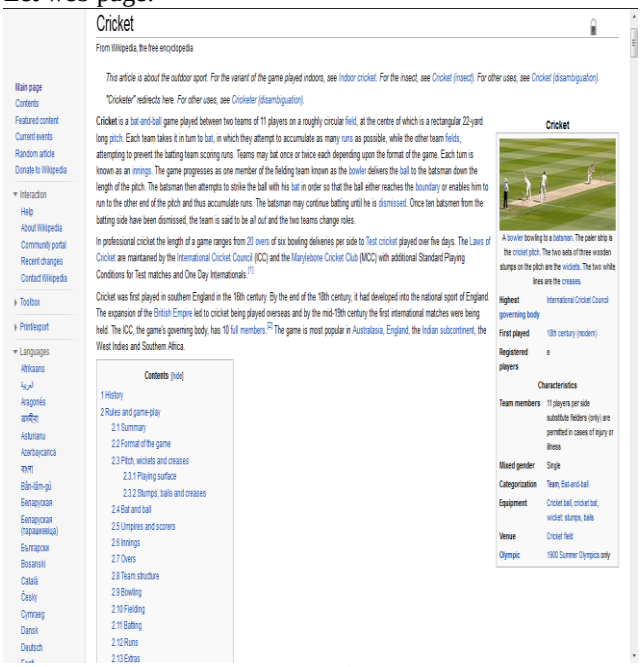

Fig. 3 Example of Web page

After pre-processing the above text found the following Frequent or Keywords words:
{cricket, ground, match, ball, bowler, pitch, catch, over, team, run, wicket }

B. Represent the Keywords in Binary Value:

When execute the apriori algorithm then generated the sets of associated words file, configuration file and transaction file and the probability file. Here below to see the transaction file. In this file 1 value assign the present of keyword and the 0 value assign the absent keyword.

Keywords Represent in Binary Value

1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
1	0	1	1	0	0	0	1	1	1	0	1	0	0	1	0	0	1	0	0	0	0	0
1	0	1	1	1	1	0	1	1	1	1	1	1	1	1	0	0	1	1	0	0	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	1
1	0	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

In this work, numbers of web pages are used as training data for learning to classify text from all five categories, for all data set are take as 50% total amount of data, which 50 are from Cricket, 41 are from Hockey, 46 are from Tennis, 43 are from Football, 44 are from Baseball. After preprocessing the text data association rule mining is applied to the set of transaction data where each frequent word set from each abstract is considered as a single transaction.

C. Deriving Associated Word sets

Each webpage is considered as a transaction in the text data. After pre-processing the text data association rule mining [5][6] is applied to the set of transaction data where each frequent word set from each webpage is considered as a single transaction. Using these transactions, to generated a list of maximum length sets applying the Apriori algorithm [5][6][8]. The support and confidence is set to 0.55 and 0.75 respectively.

TABLE II
Word set with Occurrence Frequency for 50% data sets

Word sets Found	**Number of Occurrence Documents**				
	Cricket	**Hockey**	**Tennis**	**Football**	**Baseball**
1	42				
22	37				
1 22	31				
2		40			
4			30		
9			28		
18			36		
17				29	
5					37
6					37
17					34
5 6					37
5 17					28
6 17					28
5 6 17					28

D. Associated word Set with Probability Value

To use the Naïve Bayes classifier for probability calculation the generated associated sets are required. The calculation of Equation (3), this classifier is based on the fraction of each target class in the training data. From the generated word set after applying association mining on 50% training data, and found the following information based on the result.

Total No. of Word Set = 15
Total No. of Word Set from Cricket = 3
Total No. of Word Set from Hockey = 1
Total No. of Word Set from Tennis = 3
Total No. of Word Set from Football = 1
Total No. of Word Set from Baseball = 7

Prior probability for Cricket, Hockey, Tennis, Football, and Baseball are 0.2, 0.06, 0.2, 0.06, and 0.46 respectively. Then Equation (4) is calculated according to the equation. The probability values of word set are listed in Table III.

TABLE III
Word set with Probability Value for 50% data sets

Word sets Found	Probability				
	Cricket	Hockey	Tennis	Football	Baseball
1	7.166667				
22	6.333334				
1 22	5.333334				
2		20.5			
4			5.166667		
9			4.833334		
18			6.166667		
17				1.5	
5					2.714286
6					2.714286
17					2.5
5 6					2.714286
5 17					2.071429
6 17					2.071429
5 6 17					2.714286

TABLE IV
Accuracy Regarding Different Set of Test Data

Data set							Accuracy						
							Accurate Amount of Data					Total Amount of Data Found Accurate	% of Accuracy
Total Data set %	Total Amount of Data	Cricket	Hockey	Tennis	Football	Baseball	Cricket	Hockey	Tennis	Football	Baseball		
10	44	10	8	9	8	9	7	4	5	4	0	21	45.91
15	67	15	12	14	13	13	10	0	8	8	8	34	49.56
20	88	20	16	18	17	17	14	0	9	10	11	44	57.82
25	111	25	20	23	21	22	17	12	13	11	13	66	59.17
30	134	30	25	27	26	26	20	18	18	15	17	88	65.84
35	156	35	29	32	30	30	24	22	16	16	19	97	62.47
40	179	40	33	37	34	35	28	19	20	20	28	115	61.62
45	201	45	37	41	39	39	31	30	23	23	23	130	64.39
50	224	50	41	46	43	44	37	33	29	25	29	153	67.88
55	247	55	45	51	47	49	36	36	28	24	31	155	63.03

E. Comparative Study:

In this section, I tried to represent comparative presentation in different point of views. Below Table V describe the old proposed method [5] with new proposed method with their different data set in percentage. Then after, plot for that data set.

TABLE V

Percentage of Accuracy Vs Percentage of Data sets

% of Data sets	% of Accuracy	
	Old Proposed Work	New Proposed Method
10	31	45.91
15	34	49.56
20	32	57.82
25	54	59.17
30	57	65.84
35	70	62.47
40	68	61.62
45	68	64.39
50	80	67.88
55	68	63.03

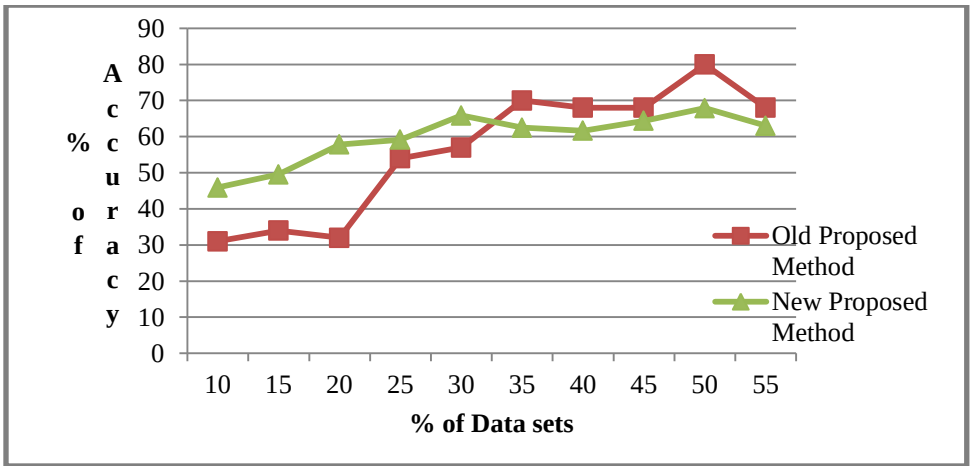

Fig. 4 % of Data sets Vs % of Accuracy

At the beginning of the experiment started with 10% of the data sets, which showed unsatisfactory accuracy. Then to increased data set to 20% which showed development in accuracy. Next as to increase the percentage of training data set accuracy became more desirable. I checked up to 55% training data. In this process, considering accuracy overall 68% accuracy i.e., 50% data set as the best.

V. Conclusion

This technique presented an efficient technique for web page classification. This technique will be more effective is the training set is set in such a way that it generates more sets. Though the experimental results are quite encouraging, it would better if the work with larger data sets with more classes. The existing technique requires more or less data for training as well as less computational time of these techniques.

VI. Future Work

In training set of data, although all the web pages have almost equal size of length, they have slightly different number of frequent words after pre-processing them. In order to avoid null attribute value in any transaction in the set of transaction database. These word sets containing null values have no use in classification. Increase the number of different types of class value for generating associated word sets. In future, take different values of support and confidence and to obtain different types of result of their classes.

References

1. Qasem A. Al-Radaideh, Eman Al Nagi "Using Data Mining Techniques to Build a Classification Model for Predicting Employees Performance", International Journal of Advanced Computer Science and Applications(IJACSA), Vol. 3, No. 2, 2012
2. Thair Nu Phyu "Survey of Classification Techniques in Data Mining", Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol I IMECS 2009, March 18 - 20, 2009, Hong Kong
3. Surjeet Kumar Yadav, Saurabh Pal " Data Mining: A Prediction for Performance Improvement of Engineering Students using Classification", World of Computer Science and Information Technology Journal (WCSIT), ISSN: 2221-0741, Vol. 2, No. 2, 51-56, 2012
4. Vladimir Gorodetsky, Oleg Karsaeyv, Vladimir Samoilov, "Multi-agent Technology for Distributed Data Mining and Classification", Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology (IAT'03) 0-7695-1931-8/03 $ 17.00 © 2003 IEEE

5. S. M. Kamruzzaman, Farhana Haider, Ahmed Ryadh Hasan "Text Classification Using Data Minig", ICTM 2005.
6. XindongWu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, Dan Steinberg "Top 10 algorithms in Data Mining", Knowl Inf Syst (2008) 14:1–37 DOI 10.1007/s10115-007-0114-2
7. Erhard Rahm, Hong Hai Do, "Data Cleaning: Problems and Current Approaches", IEEE Data Eng. Bull. 23(4):3--13 (2000), University of Leipzig, Germany http://dbs.uni-leipzig.de
8. Chowdhury Mofizur Rahman and Ferdous Ahmed Sohel and Parvez Naushad and S M Kamruzzaman, " Text Classification using the Concept of Association Rule of Data Mining", CoRR (2010) http://arxiv.org/ftp/arxiv/papers/1009/1009.4582.pdf
9. Jiawei Han and Micheline Kamber, "Data Mining: Concepts and Techniques", Second Edition Elsevier Inc.
10. Rajan Chattamvelli, "Data Mining Methods: Concepts and Applications", Narosa Publishing House Pvt. ltd. ISBN: 978-81-7319-967
11. http://www.iasri.res.in/ebook/win_school_aa/notes/Data_Preprocessing.pdf (Last visited 24 Jan 2013)
12. Springer.Web.Data.Mining.Dec.2006.pdf (Last visited 13 March 2013)
13. http://www.ijarcsee.org/index.php/IJARCSEE/article/viewFile/335/301 (Last visited 16 March 2013)